Departamento de
Informática e Ingeniería
de Sistemas
Universidad Zaragoza
1542

Trabajo Fin de Máster

Reconocimiento visual de imágenes de endoscopia con Deep Learning

Visual recognition with Deep Learning for endoscopic data

Clara Tomasini

Directora: Ana Cristina Murillo Arnal

Co-director: Luis Miguel Riazuelo Latas

Máster en Ingeniería biomédica
Departamento de Informática e Ingeniería de Sistemas
Escuela de Ingeniería y Arquitectura
Universidad de Zaragoza

22 de Junio de 2021

Resumen

La recopilación y el análisis de imágenes son una pieza fundamental dentro de los procesos de diagnóstico médico. Si bien siempre lo han sido, las aplicaciones de técnicas de *Machine Learning* en el ámbito de la medicina añaden agilidad y automatización a los procesos, permitiendo la diagnosis precoz. Este trabajo se centra en el procesado de imágenes de endoscopia mediante estas técnicas. Se trabaja en técnicas para la segmentación o clasificación de zonas de interés en las imágenes, por ejemplo reconstrucción 3D de la parte del cuerpo que aparece en esta imagen para detección y diagnóstico de enfermedades. En los últimos años se están proponiendo modelos de *Machine Learning* para el procesado de imágenes de endoscopia cada vez más precisos y eficientes, pero las imágenes de endoscopia presentan aun un reto que dificulta la generalización de los métodos de procesado: puede existir una gran variabilidad entre los imágenes debida a las condiciones particulares en las que se graban los vídeos de endoscopia. Esta variabilidad hace necesario adaptar los modelos a las imágenes particulares con las que se quiere trabajar para obtener mejores resultados.

El objetivo de este trabajo es conseguir los modelos más adecuados y mejor adaptados para preprocesar las imágenes de un nuevo dataset muy amplio capturado en el marco de un proyecto Europeo. Las tareas realizadas para conseguirlo se han agrupado en dos bloques:

Se han estudiado técnicas del estado del arte supervisadas para **segmentación de herramientas en imágenes de endoscopia**. Además, se han estudiado técnicas eficientes en casos más generales de segmentación semántica. Se han re-entrenado los métodos existentes con datos de endoscopia reales (un dataset publicado para un challenge de segmentación de herramientas de endoscopia; otro dataset propio del proyecto en el que se desarrolla este trabajo). Se ha hecho el fine-tuning de las técnicas ya existentes para endoscopia utilizando el dataset propio del proyecto. Se ha hecho entrenado desde cero el modelo más eficiente con el dataset público, para adaptarlo a imaágenes de endoscopia y luego el fine-tuning de este mismo modelo con el dataset del proyecto. Se han evaluado todos estos modelos antes y después del re-entrenamiento (fine-tuning) con los dos datasets descritos previamente.

También se han estudiado algunas **técnicas no supervisadas para análisis de datos**, en particular para reducir dimensionalidad y poder visualizar de manera más adecuada un resumen de conjuntos de datos muy grandes utilizando características extraídas por los modelos de segmentación previamente estudiados para describir estos datos. Se ha utilizado el método PCA para reducción de dimensionalidad, y el método t-SNE para análisis y visualización del contenido de vídeos de endoscopia. En particular se ha analizado la distribución de los datos en las visualizaciones obtenidas después de aplicar estos dos métodos a los descriptores de los datos.

Como conclusiones principales, se puede notar primero que el re-entrenamiento hecho de los modelos ya existentes de segmentación permiten efectivamente obtener modelos adaptados a las imágenes del proyecto con valores de las métricas de evaluación parecidos a los del estado del arte. Se puede también notar que las visualizaciones obtenidas con las técnicas no supervisadas muestran que los descriptores extraídos por estos mismos modelos de segmentación permiten agrupar los datos según criterios sobre el contenido semántico de cada frame.

Índice general

Capítulo 1

Introducción

Este capítulo presenta la motivación del trabajo y el contexto en el que se ha desarrollado, así como un resumen de sus objetivos y su contenido.

1.1. Motivación

En los últimos años, se observa como la inteligencia artificial y en particular el campo del *Machine Learning*, están teniendo numerosas aplicaciones en el ámbito de la medicina, con el objetivo de ayudar y mejorar el trabajo ya realizado por especialistas. Así, las técnicas de *Machine Learning* se pueden utilizar para desarrollar herramientas de asistencia al diagnóstico que pueden ayudar con el diagnóstico precoz y la prevención de enfermedades, como por ejemplo herramientas de ayuda al diagnóstico de cáncer de mama mediante análisis de imágenes de histología para detectar tejidos y células cancerosas [1], o de ayuda al diagnóstico de osteoartritis en rodilla a partir de imágenes de radiografía [2], así como herramientas de reconstrucción 3D de partes del cuerpo humano para obtener más información sobre la estructura de esta zona, por ejemplo para detectar y localizar de forma más precisa pequeños tumores en el intestino [3].

Este trabajo se centra en el desarrollo de una herramienta de ayuda en el procesado de imágenes de endoscopia. La endoscopia es una técnica clave para cirugía mínimamente invasiva. Permite a un médico visualizar una cavidad o el interior de un órgano del paciente mediante una cámara que se inserta en una apertura del cuerpo. Es un procedimiento muy frecuente, que se puede utilizar en distintas partes del cuerpo, como por ejemplo el estómago (gastroscopia), el colon (colonoscopia), los bronquios (broncoscopia) o el abdomen (laparoscopia). Se puede ver un ejemplo de imagen de colonoscopia en la figura 1.1.

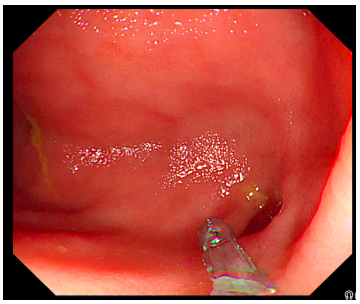


Figura 1.1: Ejemplo de imagen de colonoscopia

En general, los procedimientos de endoscopia se suelen utilizar para detectar posibles lesiones o enfermedades mediante el análisis de los vídeos por especialistas. La visión por computador puede ayudar a mejorar estos procedimientos mediante un procesado automático de los vídeos de endoscopia, que puede servir de ayuda al médico para realizar el diagnóstico. Estos dos análisis paralelos de los datos pueden permitir un diagnóstico mas precoz que teniendo solo el análisis del médico. Uno de los primeros pasos para automatizar tareas de procesado de estos vídeos es la comprensión automática de los elementos presentes en la escena. Estos elementos pueden ser por ejemplo herramientas de cirugía, como aparece en la figura 1.1, o elementos característicos de la anatomía de la zona en la que se realiza la endoscopia, como por ejemplo los pliegues del intestino en el caso de gastroscopia. La figura 1.2 presenta el resultado del procesado para detección de herramientas en vídeos de colonoscopia. En los últimos años, este tipo de aplicaciones están teniendo gran impulso gracias a técnicas de "deep learning", que se discuten en más detalle en el capitulo 2. En este TFM se propone estudiar técnicas supervisadas y no-supervisadas para el procesado automático de los vídeos de endoscopia, como se detalla en las siguientes secciones de este capitulo.

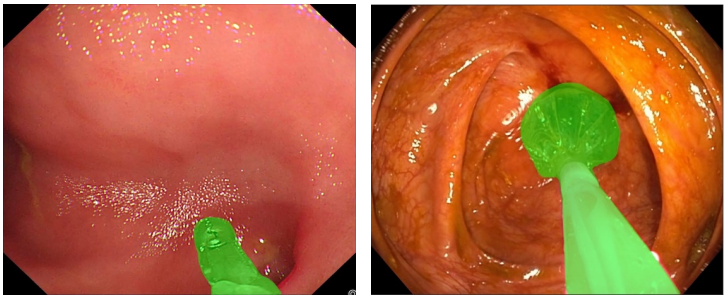


Figura 1.2: Ejemplos de imágenes de colonoscopia con herramientas segmentadas

1.2. Contexto

Este proyecto se ha desarrollado en el ámbito del proyecto EndoMapper, un proyecto europeo de investigación, coordinado desde el grupo de investigación

de Robótica, Percepción y Tiempo Real (RoPeRT) del Instituto Universitario de Investigación en Ingeniería de Aragón (I3A).

El proyecto EndoMapper [4] es un proyecto de investigación biomédica que tiene como objetivo realizar modelos 3D del interior del cuerpo humano a partir de imágenes de endoscopia utlizando técnicas de visión por computador y *Machine Learning*. Estos modelos pretenden facilitar la tarea del personal médico tanto durante los procedimientos como en un posterior seguimiento de los pacientes a lo largo del tiempo.

1.3. Tareas y Objetivos

El objetivo general de este trabajo es el desarrollo de técnicas de aprendizaje automático para detección de distintos elementos de interés en las imágenes de endoscopia disponibles en el proyecto EndoMapper.

Los objetivos más concretos son:

- Conseguir un sistema automático para segmentación de herramientas médicas en las imágenes de endoscopia. Esto es necesario para poder separar automáticamente qué partes de las imágenes de los vídeos pertenecen a elementos "externos" a la escena real que se está analizando.
- Conseguir un sistema que identifique de manera automática nuevos elementos de interés dentro de dichas secuencias de datos, por ejemplo, zonas específicas de la topología del intestino, que tienen interés médico, como el ciego, o artefactos que aparecen en algunos frames y hacen necesario eliminar dichos frames de las secuencias.

Las principales tareas planteadas para alcanzar los objetivos descritos son las siguientes (su distribución temporal se puede ver en el cronograma de la tabla 1.1):

- **Tarea 1.** Estudiar el estado del arte en modelos para segmentación de herramientas médicas en las imágenes de endoscopia, en particular técnicas recientes basadas en Deep Learning y sus fundamentos
- **Tarea 2.** Adaptar y evaluar dichos modelos con métricas estandar para el tipo de imagen de endoscopia a utilizar en el proyecto
- **Tarea 3.** Estudiar y ejecutar técnicas no supervisadas para descubrimiento de patrones y elementos de interés en secuencias
- **Tarea 4.** Diseñar, implementar y evaluar modelos que permitan detectar dichos elementos automáticamente en secuencias reales de endoscopia
- **Tarea 5.** Escribir documentación e informes técnicos sobre los métodos o adaptaciones propuestas y los resultados obtenidos

	SEP.	OCT.	NOV.	DIC.	ENE.	FEB.	MAR.	ABR.	MAY.	JUN.
Tarea 1										
Tarea 2										
Tarea 3										
Tarea 4										
Tarea 5										

Tabla 1.1: Este cronograma presenta la distribución temporal aproximada de las tareas del proyecto.

1.4. Estructura de la memoria

Este documento está organizado en los siguientes capítulos. El capítulo 2 presenta el estado del arte a partir del que se ha realizado el proyecto. Los capítulos 3 y 4 recopilan el trabajo realizado para alcanzar los objetivos planteados. El capítulo 5 presenta los experimentos realizados a partir de las soluciones propuestas y sus resultados, y el capítulo 6 resume las conclusiones que se pueden sacar de este trabajo.

Capítulo 2

Aprendizaje automático en endoscopia

En los últimos años, se han desarrollado muchas aplicaciones de técnicas de aprendizaje automático, *Machine Learning*, en muchos ámbitos de la medicina y en concreto también, de particular interés en este trabajo, para endoscopia [5]. Esta técnicas se utilizan como ayuda para mejorar los resultados del análisis de imágenes de endoscopia. Así, existen modelos que permiten hacer reconocimiento y clasificación de zonas anatómicas [6], diagnóstico de enfermedades como cáncer gástrico mediante detección y clasificación de lesiones [7], o detección, segmentación y clasificación de pólipos en el caso de colonoscopia [8].

En el caso del proyecto Endomapper, dentro del que se desarrolla este trabajo, el objetivo es poder construir en tiempo real modelos 3D del interior del cuerpo humano. Esta reconstrucción y el hecho de poder localizar la cámara en todo momento dentro de este modelo, pueden facilitar tareas de re-localización de lesiones, revisiones o un análisis automatizado de las exploraciones realizadas. Para poder hacer esta reconstrucción, es necesario procesar primero las imágenes para detectar y localizar los elementos de interés para realizar la reconstrucción, por ejemplo, puntos característicos que pertenecen al cuerpo, así como los elementos ajenos al cuerpo, como por ejemplo las herramientas utilizadas durante el procedimiento de endoscopia, ya que hay que filtrarlas a la hora de realizar las reconstrucciones.

En este trabajo se utilizan técnicas de aprendizaje automático supervisado para segmentar las herramientas y no supervisado para intentar encontrar otras zonas de interés.

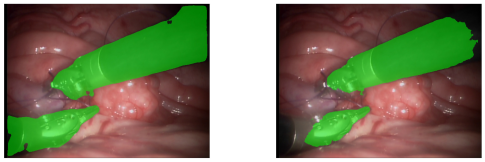
2.1. Segmentación de herramientas con modelos supervisados

Un procesado habitual en los vídeos de endoscopia es la segmentación de las herramientas de endoscopia para poder detectar su presencia y determinar su posición y trayectoria [9]. Este procesado consiste en un problema de segmentación binaria necesario para poder separar automáticamente las partes de los vídeos que pertenecen a elementos "externos" a la escena real que se está

analizando, como las herramientas quirúrgicas.

Ya existen modelos de segmentación de herramientas en imágenes de endoscopia desarrollados para el *challenge* EndoVisSub 2017 [10]. El objetivo de este *challenge* era obtener un modelo de segmentación semántica binaria de herramientas en vídeos de endoscopia, es decir, etiquetar cada píxel de una imagen de endoscopia según si pertenece a la herramienta o no. La solución ganadora del *challenge* [11] evalúa 4 modelos de segmentación: U-Net [12], 2 versiones de TernausNet [13] y una versión de LinkNet [14]. Como se puede ver en la figura 2.1, estos modelos entrenados para segmentar las imágenes de endoscopia dan segmentaciones bastante correctas. No obstante, no son los más eficientes, ya que tardan unos 100-200ms en calcular cada máscara de segmentación, lo que puede impedir su uso en tiempo real.

(a) Imagen original (b) U-Net (c) TernausNet-11



(d) TernausNet-16 (e) LinkNet-34

Figura 2.1: Segmentaciones obtenidas con los modelos entrenados para el challenge EndoVisSub 2017 por el equipo ganador.

Para obtener un modelo de segmentación adaptado a las imágenes del proyecto y más eficiente, se ha considerado entrenar un nuevo modelo desde cero pero utilizando alguna arquitectura de red más eficiente. Existen muchas propuestas recientes, como explicado en [15], para conseguir modelos de segmentación semántica eficientes que se puedan usar en tiempo real y sin necesitar un volumen de memoria muy alto, es decir que se puedan ejecutar en CPU. Uno de estos modelos más eficientes es MiniNet [16], desarrollado por el grupo de investigación dentro del que se ha realizado este trabajo. Se trata de un modelo de segmentación semántica desarrollado con el objetivo de ser lo más eficiente posible en términos de memoria y tiempo de inferencia, lo que permite usarlo en tiempo real y en sistemas sin muy alta capacidad de cómputo.

2.2. Análisis automático de vídeos de endoscopia con métodos no supervisados

Otro procesado interesante en los vídeos de endoscopia es la detección y la clasificación automática de eventos o zonas característicos del vídeo. Pueden ser por ejemplo las distintas partes del sistema digestivo en el caso de colonoscopia que permiten localizar de manera más precisa cada píxel de la imagen y permitir así una reconstrucción 3D mejor del cuerpo. Pueden ser también lesiones y tumores [17], con el objetivo de diagnosticar cánceres antes y así aumentar la probabilidad de que el tratamiento funcione.

Cuando hay datos de ejemplo suficientes, se puede entrenar un modelo de clasificación supervisado utilizando imágenes de distintas clases, todas anotadas, pero sino, resulta interesante intentar descubrir patrones salientes a lo largo de la endoscopia de manera no supervisada, es decir utilizando técnicas no supervisadas aplicadas a las imágenes sin anotar.

Estas técnicas permiten agrupar imágenes con características comunes y así hacer un procesado automático del vídeo que permite destacar las partes del vídeo con posibles puntos de interés, sean lesiones o zonas anatómicas específicas. Este procesado puede también ser una forma de obtener un pseudo-etiquetado automático de conjuntos muy grandes de datos [18], para poder utilizarlos luego en otros procesos.

Para estas técnicas no supervisadas, resulta esencial tener primero una representación compacta de cada frame, y luego analizar que patrones se detectan para saber si los descriptores (*features*) usados para obtener la representación de cada frame son adaptados para agrupar los distintos frames en clases.

Una manera de obtener los mapas de características es utilizando como *feature extractor* modelos [19] ya entrenados en imágenes de endoscopia, por ejemplo los modelos utilizados para segmentar las herramientas. Una vez obtenidos los mapas de características que permiten representar cada uno de los frames, se pueden aplicar por ejemplo técnicas de *clustering* [19] o técnicas de reducción de dimensionalidad como t-SNE [20] para visualizar los descriptores y su distribución en un espacio de menos dimensiones y así comprobar si agrupan las imágenes con algún criterio interesante. Se puede también aplicar a estos mapas de características un clasificador SVM [17] entrenado en datos etiquetados para poder separar los frames en distintas clases.

Capítulo 3

Modelos supervisados para segmentación de herramientas

En este capítulo se presentan las arquitecturas de los distintos modelos supervisados utilizados así como el procedimiento de trabajo seguido para obtener modelos de segmentación adaptados a las imágenes propias del proyecto.

3.1. Arquitecturas de red utilizadas

La parte de segmentación de herramientas se hace utilizando 4 modelos de aprendizaje supervisado, es decir que utilizan datos de entrenamiento etiquetados. Tres de estos modelos son los desarrollados en la solución ganadora [11] del challenge EndoVisSub2017 [10]: U-Net, TernausNet-11 y LinkNet-34. El otro modelo es MiniNet, desarrollado por el grupo de investigación RoPeRT dentro del que se realiza este trabajo.

3.1.1. U-Net

El modelo U-Net [12] es una red neuronal convolucional diseñada para obtener mejores resultados en tareas de segmentación de imágenes médicas, permitiendo asignar a cada píxel de la imagen una etiqueta de clase, en vez de asignar una etiqueta única a toda la imagen, lo que es más frecuente en tareas de clasificación de imágenes. La arquitectura de este modelo, presentada en la figura 3.1, se compone de dos partes simétricas :

- El codificador o camino de contraccción, formado por una serie de capas de convolución que permiten hacer un submuestreo de la imagen y sacar así información cada vez más detallada de esta imagen, aumentando cada vez más el número de *feature channels*.
- El decodificador o camino de expansión, formado por capas de convolución y deconvolución que permiten reducir poco a poco el número de *feature channels* y así obtener en la última capa de salida una máscara de segmentación del mismo tamaño que la imagen de entrada.

El alto número de *feature channels*, provenientes de las diferentes capas del codificador y concatenadas con las *feature channels* del decodificador, que se

conservan en el decodificador son lo que permite conservar más información de contexto y así poder asignar una etiqueta a cada píxel.

Esta arquitectura, combinada con el uso de *data augmentation*, también permite reducir el número de imágenes anotadas necesarias para entrenar el modelo y conseguir buenos resultados.

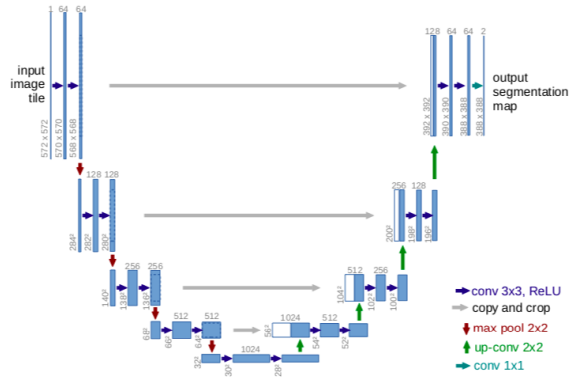


Figura 3.1: Arquitectura del modelo U-Net [12]

3.1.2. TernausNet-11

El modelo TernausNet-11 [13] es una variación del modelo U-Net que reemplaza el codificador original por el codificador VGG11 pre-entrenado en otro dataset. La arquitectura del modelo se puede ver en la figura 3.2.

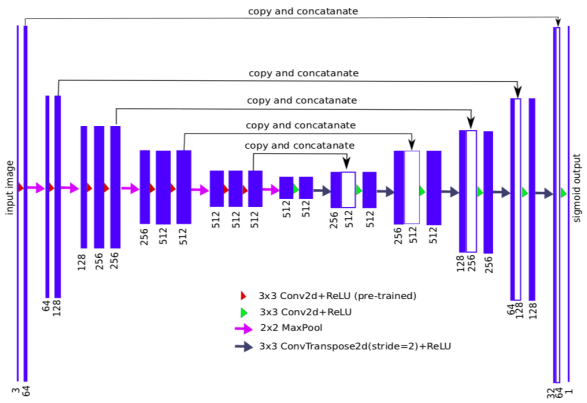


Figura 3.2: Arquitectura del modelo TernausNet-11 [13]

El uso del codificador pre-entrenado permite mejorar los resultados obtenidos con U-Net sin necesitar más tiempo ni imágenes para entrenar el modelo.

3.1.3. LinkNet34

El modelo LinkNet34 [11, 14] sigue el mismo principio que el modelo TernausNet-11 pero utiliza ResNet34 preentrenado como codificador. Su estructura se puede ver en la figura 3.3. El uso de ResNet34 en vez de VGG11 permite obtener resultados parecidos pero de forma más rápida en las mismas imágenes.

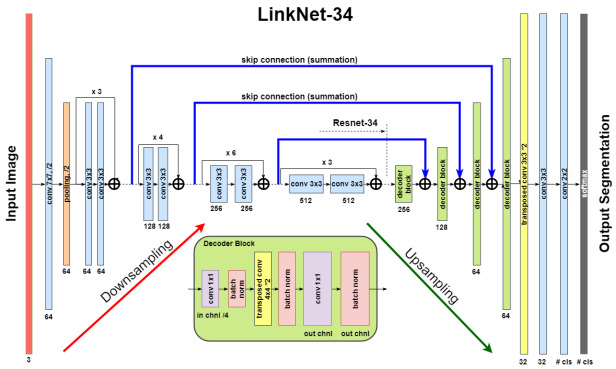


Figura 3.3: Arquitectura del modelo LinkNet34 [11]

3.1.4. MiniNet

El modelo MiniNet [16] se compone de 4 bloques que se pueden ver en la figura 3.4. El primer bloque y el último (*Downsample module* y *Upsample module*) permiten, al igual que en el caso del modelo U-Net, hacer primero un submuestreo de la imagen para aumentar el número de *feature channels* y luego reducir este número de *feature channels* para obtener una máscara de segmentación del mismo tamaño que la imagen original. Estos dos bloques están formados por capas sucesivas de convolución. El segundo y el tercer bloques son los que permiten extraer la información de la imagen con distintos niveles de precisión. Este modelo está desarrollado con el objetivo de ser lo más eficiente posible en términos de memoria y tiempo de inferencia, lo que se obtiene mediante el uso de las distintas capas de convolución en los cuatro bloques. Obtiene buenos resultados en imágenes de ciudades, calles y coches, pero no está entrenado ni desarrollado para utilizarlo específicamente en imágenes médicas.

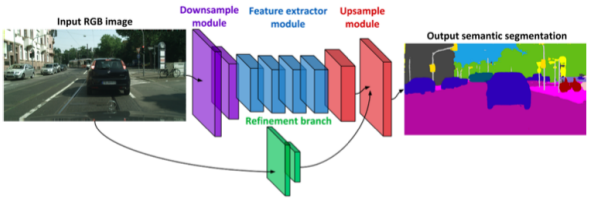


Figura 3.4: Arquitectura del modelo MiniNet [16]

3.2. Sistema de segmentación implementado

El procedimiento de trabajo seguido para obtener un modelo de segmentación adaptado a los datos del proyecto está basado en técnicas de *transfer learning*. Es decir, construir modelos adaptados a partir de los modelos descritos en la parte anterior.El proceso se puede ver resumido en la figura 3.5. Consiste en hacer el *fine-tuning* de cada modelo utilizando datos del proyecto anotados con *ground truth*.

El *fine-tuning* es una técnica habitual de *transfer learning*. Consiste en entrenar más un modelo existente ya entrenado en un dominio para adaptarlo a otro relacionado. Permite así mejorar los resultados obtenidos con este modelo en el nuevo dominio utilizando menos datos anotados de entrenamiento que entrenándolo desde cero.

En particular, el procedimiento seguido para todos los modelos entrenados consiste en las siguientes dos fases:

Obtención de modelos iniciales. Los modelos iniciales del sistema implementado son los 4 modelos presentados en la parte anterior: UNet, TernausNet-11, LinkNet-34 y MiniNet. De estos modelos, UNet, TernausNet-11 y LinkNet-34 ya estaban disponibles entrenados en imágenes de endoscopia en [11] para el Challenge EndoVisSub2017. Por el contrario, el modelo MiniNet se ha tenido que entrenar desde cero ya que no estaba disponible pre-entrenado en imágenes de endoscopia. Se ha entrenado en un conjunto de imágenes de endoscopia parecidas a las usadas en [11] para entrenar los otros 3 modelos, de forma que la segunda fase se aplique a modelos todos pre-entrenados en imágenes de endoscopia.

Adaptación de los modelos: Fine-tuning. La segunda fase es la fase de *fine-tuning*. Para hacer el *fine-tuning* se utiliza el código de entrenamiento disponible para cada uno de los modelos y se adapta para poder utilizarlo con los modelos pre-entrenados, de forma que, al lanzar este código de entrenamiento, no se cargue solo el modelo sino también los pesos del pre-entrenamiento para inicializar el modelo. Como detalle más práctico, también se ha adaptado la manera de cargar las imágenes del proyecto, ya que los distintos conjuntos de *train* y *test* no están estructurados de la misma manera que para el pre-entrenamiento hecho en [11]. La estructura del código desarrollado para este trabajo se puede ver en el apéndice B.

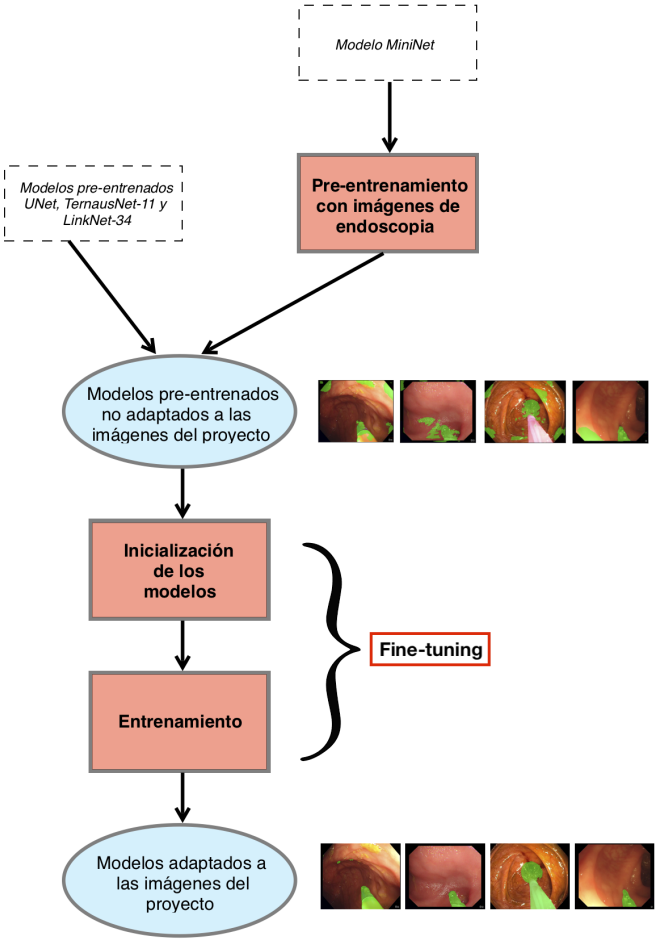


Figura 3.5: Procedimiento de trabajo seguido

Capítulo 4

Técnicas no supervisadas de análisis automático de datos

En este capítulo se presenta un método no supervisado basado en las características aprendidas por los modelos de segmentación y que permite analizar de manera automática los frames de un vídeo de endoscopia. Se presentan a continuación las distintas etapas implementadas para desarrollar este método.

4.1. Metodología de trabajo

El objetivo de esta parte es comprobar si las características aprendidas por los modelos de segmentación adaptados en la parte 3 permiten agrupar los frames en varios grupos según características comunes. El método desarrollado para llegar a este objetivo se puede dividir en 3 etapas. Su estructura se resume en la figura 4.1 y se detalla en las siguientes secciones.

- La primera etapa consiste en extraer de los frames las características que permiten representar cada frame. Para extraer estas características se utilizan los modelos de segmentación adaptados en la primera parte de este trabajo.
- La segunda etapa es el procesado de los mapas de características extraídos para permitir su visualización. En efecto, los mapas extraídos son de dimensionalidad muy alta, por lo que es necesario reducir primero esta dimensionalidad para facilitar su visualización. De esta forma, el número de datos a visualizar es menor pero no se pierde información importante.
- Una vez estén reducidas las dimensiones de los mapas de características, se pueden visualizar los datos mediante técnicas de visualización. Estas técnicas permiten visualizar datos de dimensionalidad relativamente alta en un espacio de 2 o 3 dimensiones en el que se puede hacer una interpretación visual de los resultados. Así, a partir de las visualizaciones obtenidas se puede comprobar si las características extraídas para describir los datos permiten diferenciar entre los distintos frames según si los puntos que representan cada frame aparecen agrupados o no.

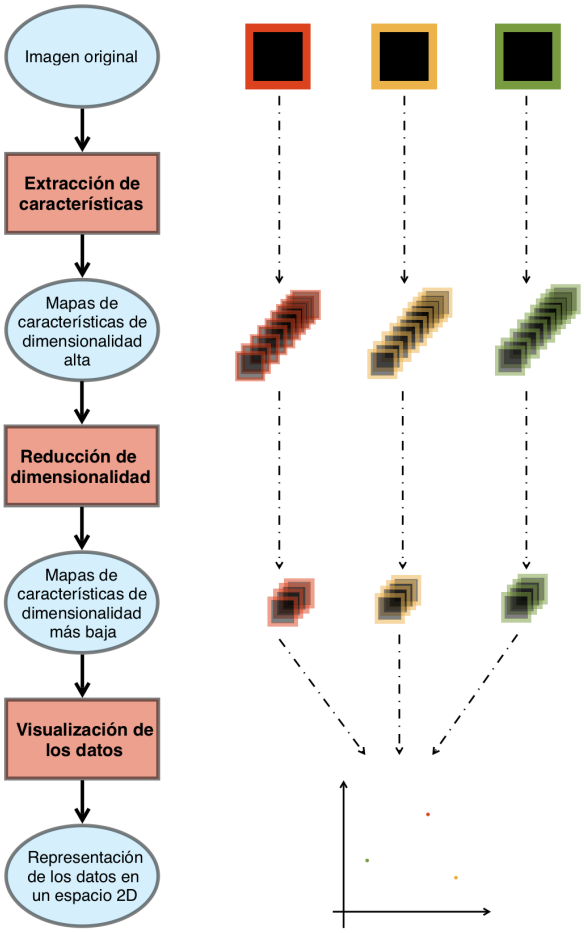


Figura 4.1: Estructura del método desarrollado

4.2. Implementación

4.2.1. Extracción de características

Una forma de extraer las características de cada frame es utilizar una red neuronal pre-entrenada en datos parecidos a los que se quieren procesar, como explicado en [21]. En el caso de este trabajo, se utilizan los modelos de segmentación de herramientas ya adaptados a los datos propios del proyecto. Para obtener un *feature extractor* a partir de estos modelos, se conserva solo la parte de extracción de características (camino de contracción como aparece en el capítulo 3), sin la parte que permite obtener la máscara de segmentación binaria (camino de expansión). Con esta parte del modelo, se obtienen mapas de características de tamaño muy alto (2703360 características con MiniNet, 675840 con LinkNet34) para cada uno de los frames del vídeo.

4.2.2. Reducción de dimensionalidad

El número muy alto de características extraídas por el modelo para cada frame implica la presencia de información redundante y de poco interés, lo que impide su uso directo para separar los frames en grupos de manera eficiente. Por eso, es necesario reducir las dimensiones de los mapas de características para conservar solo la información más importante. Un método posible de reducción de dimensiones es el análisis en componentes principales (PCA) [22]. Se trata de un método estadístico que permite expresar un conjunto grande de datos utilizando un número determinado de componentes o ejes, cada uno de los ejes permitiendo expresar características más o menos detallada o relevante. De esta forma, se reduce el tamaño de los datos que representan cada uno de los frames sin perder información importante. La figura 4.2 presenta el efecto del PCA aplicado a datos en un espacio 3D. En este caso, cada punto pasa de estar representado por coordenadas en 3D a coordenadas en 2D.

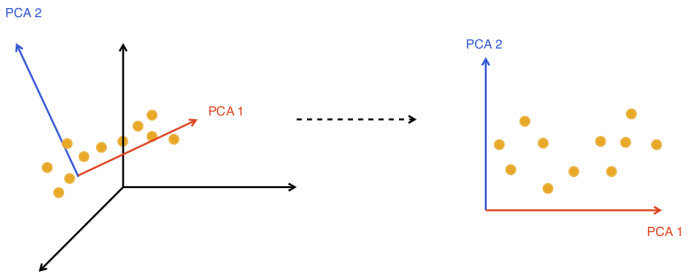


Figura 4.2: Efecto del PCA al reducir una seria de puntos de 3D a 2D

En el caso de este trabajo, la idea será pasar de 703360 características con MiniNet y 675840 con LinkNet34 a un número más manejable que se pueda procesar por el t-SNE, en este caso 50.

4.2.3. Visualización de los datos

A partir de las componentes obtenidas con el PCA, se pueden visualizar los datos para comprobar cualitativamente si las características extraídas utilizando los modelos de segmentación permiten o no separar los frames en categorías. Si permiten separar los frames, se podrían utilizar entonces para desarrollar herramientas de clustering y posterior clasificación.

Un método posible para visualizar estos datos es el t-SNE [20]. Se trata de una técnica de reducción de dimensiones basada en probabilidades que permite visualizar datos con alta dimensionalidad en un espacio de 2 o 3 dimensiones. En el caso de este trabajo, el uso de t-SNE permite visualizar las imágenes de endoscopia disponibles en una misma figura 2D asignando a cada imagen (conjunto de componentes principales que la describen) coordenadas 2D para localizarla en la figura de visualización. A partir de la representación obtenida, es posible determinar si las características extraídas por los modelos de segmentación van a permitir o no diferenciar entre los distintos tipos de imágenes de interés. La figura 4.3 presenta el efecto del método t-SNE en las componentes obtenidas en 2D después del PCA.

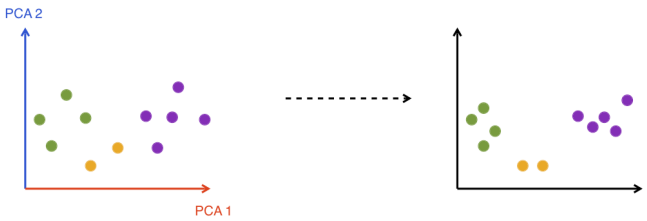


Figura 4.3: Efecto del t-SNE en 2D

Capítulo 5

Experimentación

En este capítulo se presenta la validación experimental de las técnicas de segmentación de herramientas y análisis no supervisado de vídeos presentadas en el capítulo anterior.

5.1. Entorno de experimentación

Esta sección detalla el entorno en el que se han realizado los experimentos, incluyendo datos utilizados, máquinas utilizadas para los experimentos, métricas comparadas para la evaluación de los distintos modelos y etiquetado hecho de los datos para la primera tarea de segmentación.

5.1.1. Datos utilizados

Dataset EndoVisSub 2015 (Endo15) [23]. Dataset público de imágenes de cirugía laparoscópica de colón en las cuales aparecen distintos tipos de herramientas utilizadas durante este procedimiento. Consiste en 160 imágenes anotadas, de las cuales 96 se utilizan para *train* (**Endo15-A**) y 64 para *test* (**Endo15-B**), y 140 imágenes sin anotar utilizadas para *validation*. Las máscaras de segmentación dadas son binarias con 0 si el píxel pertenece al fondo y 1 si pertenece a la herramienta.

Dataset EndoVis 2017 (Endo17) [10]. Dataset utilizado para entrenar los modelos de la solución ganadora [11] del Challenge EndoVis 2017. Contiene imágenes similares a las del dataset Endo15 pero los organizadores del Challenge no lo han dejado público.

Datos del proyecto EndoMapper (EM). Dataset compuesto de 5 vídeos (A, B, C, D y E) de colonoscopia de una duración de entre 20 y 30 minutos grabados en el Hospital Clínico Universitario Lozano Blesa de Zaragoza (vídeo C) y otros hospitales asociados al proyecto (vídeos A, B, D y E). Todos estos vídeos se han grabado durante procedimientos reales realizados en pacientes en la actividad habitual del hospital. Estos datos tienen una dificultad de procesamiento muy alta, debido a condiciones de luz poco óptimas (falta o exceso de

Vídeo	A	B	C	D	E
Imágenes anotadas	11099	2632	3168	22	511

Tabla 5.1: Imágenes etiquetadas para cada vídeo

luz), presencia de agua, contraste bajo, colores rojos/naranjas en toda la imagen y falta de nitidez en algunos frames debido al movimiento del endoscopio. En estos vídeos se han seleccionado 17400 *frames* en los cuales aparecen herramientas utilizadas en endoscopia. Para todos ellos, se ha obtenido un etiquetado preciso de la parte del frame que ocupan las herramientas, como se detalla a continuación. Algunos de estos frames se pueden ver en el apéndice A.

Etiquetado de EM. La segmentación de las herramientas en los frames seleccionados se ha etiquetado a mano, utilizando una herramienta de Odin Vision [24]. El objetivo es obtener máscaras de segmentación que pueden ser utilizadas como *etiquetas* en los entrenamientos supervisados de los distintos modelos estudiados. Como se puede ver en la figura 5.1, en cada imagen se define el contorno de la herramienta utilizando puntos para formar un polígono. Los píxeles que se encuentran dentro del polígono, marcados con 1, son píxeles de la herramienta y el resto de píxeles, marcados con 0, pertenece al resto de la imagen. La tabla 5.1 recopila el número de imágenes anotadas para cada vídeo.

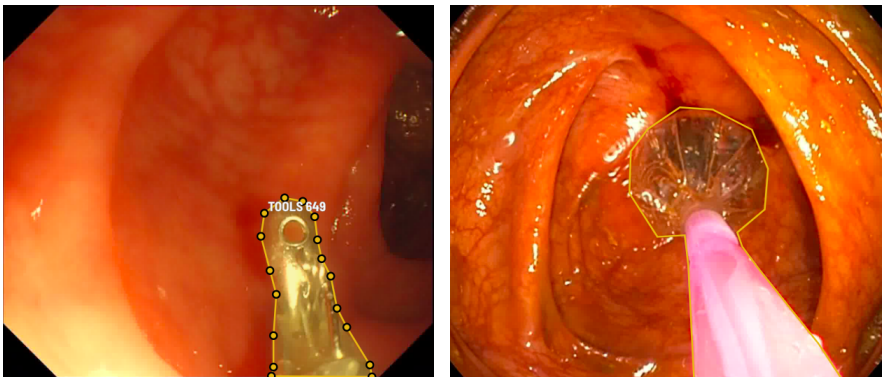


Figura 5.1: Segmentación manual de herramientas

5.1.2. Entorno de trabajo

Para los experimentos se han utilizado dos entornos diferentes:

- Google Colaboratory (Colab) [25] : Herramienta de Google que permite editar y ejecutar en el navegador código escrito en Python dentro de *notebooks* de Jupyter utilizando una GPU remota cuyo modelo (K80 o T4) es aleatorio. Permite utilizar datos (imágenes en este caso) almacenados en Google Drive.
- Ordenador de sobremesa con GPU modelo Tesla V100 SXM2, CPU Intel(R) Core(TM) i7-10700K y 64GB de memoria RAM, utilizado para

entrenar los diferentes modelos desarrollados y evaluados en el proyecto.

5.1.3. Métricas de evaluación

- ***Intersection over Union* (IoU)** : métrica más frecuente para evaluar la precisión de modelos de segmentación semántica. Se calcula con la siguiente expresión

$$IoU = \frac{\text{área de intersección}}{\text{área de unión}}, \tag{5.1}$$

  donde el *área de intersección* es los píxeles en los que la clase asignada por el modelo corresponde con la clase real definada por la etiqueta, y el *área de unión* es todos los píxeles asignados a una clase sea por la etiqueta real o por el modelo de predicción.

- **Tiempo de inferencia**: tiempo en milisegundos que tarda el modelo en calcular la máscara de segmentación de una imagen nueva

- **Número de parámetros**: número de parámetros del modelo, permite tener una idea de la memoria que ocupa el modelo

5.2. Segmentación de herramientas en vídeos de endoscopia

5.2.1. Re-entrenamiento de modelos en datos del proyecto

Objetivo. En primer lugar se quiere mostrar la eficacia del *fine-tuning*, es decir, refinar o *adaptar* modelos de segmentación de herramientas existentes con las imágenes propias del proyecto para intentar mejorar los resultados y las segmentaciones obtenidas.

Descripción. Se ha realizado el *fine-tuning* para tres de los modelos del challenge EndoVis utilizando el vídeo B del dataset EM. Como parámetros de entrenamiento 20 épocas, batch size de 4 y learning rate de 0,0001. El conjunto de *train* consiste en 1580 imágenes y el conjunto de *test* en 1053 imágenes del vídeo B, por lo que el conjunto de train representa el 60 % de los datos disponibles y el conjunto de test el 40 %. Los datos que forman cada uno de los dos conjuntos se eligen de forma aleatoria.

Resultados. La Tabla 5.2 muestra los valores de IoU obtenidos para las imágenes del dataset Endo15 y los vídeos D y E del dataset EM con los modelos antes y después del fine-tuning (marcados con el prefijo EM). De este estudio se pueden sacar varias conclusiones:

- Como se podría esperar, dado que se re-entrenan los modelos utilizando parte de las imágenes propias del proyecto, los valores de IoU en el dataset EM mejoran para todos los modelos después del fine-tuning. Así, se obtienen modelos más adecuados para segmentar herramientas en imágenes obtenidas a partir de endoscopias del proyecto. De la misma manera, los resultados de estos mismos modelos empeoran en el dataset Endo15-B, ya

que las imágenes de este dataset son distintas a las del proyecto utilizadas para el fine-tuning.

- El mejor modelo para el dataset EM, es decir el con IoU más alto, es el modelo EM-LinkNet34. Este modelo utiliza la arquitectura LinkNet34 [11] entrenado en Endo17 y fine-tuned con EM, llegando al 65.81 % en el vídeo D y 69.76 % en el vídeo E.

Modelo	**Challenges EndoVis**		**Dataset EM**	
	Dataset Endo17	**Dataset Endo15-B**	**Vídeo D**	**Vídeo E**
U-Net	75.44	20.17	12.95	06.62
EM-U-Net	/	11.49	44.33	56.19
TernausNet-11	81.14	52.21	34.39	48.79
EM-TernausNet-11	/	10.19	35.64	60.01
LinkNet34	82.36	57.36	26.50	40.59
EM-LinkNet34	/	07.44	**65.81**	**69.76**

Tabla 5.2: Valores de IoU (%) obtenidos para los distintos modelos en los datasets Endo15 y EM. Los modelos EM-* son los obtenidos después del fine-tuning con datos del proyecto (parte del video B).

La Figura 5.4 muestra ejemplos de segmentaciones obtenidas con cada uno de los modelos antes y después del fine-tuning para imágenes de Endo15 y EM. En estas imágenes se pueden comprobar cualitativamente los resultados que aparecen en la tabla anterior. Los modelos después del fine-tuning funcionan mejor que los originales en los datos del proyecto, mientras que los modelos originales dan segmentaciones mejores en imágenes de Endo15, más parecidas a las usadas para el entrenamiento del modelo original (del challenge EndoVis17). También resulta interesante evaluar la eficiencia de los modelos en cuanto a memoria y tiempo de inferencia. Esta información se puede ver en la tabla 5.3.

Modelo	**Tiempo de inferencia medio (ms)**	**Número de parámetros (Millones)**
U-Net	128	7.85
TernausNet-11	245	25.36
LinkNet34	95	21.79

Tabla 5.3: Tiempo de inferencia medio en el dataset EM y número de parámetros (en millones) de los tres modelos entrenados.

A partir de los resultados cuantitativos y cualitativos obtenidos se pueden sacar tres conclusiones :

- El fine-tuning permite adaptar mejor los modelos a los datos del dataset EM del proyecto Endomapper, lo que resulta en segmentaciones más parecidas a las segmentaciones manuales (reales) y valores de IoU más altos. En el caso del modelo LinkNet34 por ejemplo, el IoU pasa de 26.50 % en

el vídeo D y 40.59 % en el vídeo E a 65.81 % en el vídeo D y 69.76 % en el vídeo E.

- De los 3 modelos en los que se ha hecho el fine-tuning, el modelo LinkNet34 da los mejores resultados, es decir los valores de IoU más altos y las segmentaciones más correctas visualmente
- En cuanto al tiempo de inferencia y el número de parámetros, LinkNet34 también es el más rápido y, aunque tenga un número de parámetros más alto que el modelo U-Net, da mejores resultados de forma más rápida. Destaca también que TernausNet-11 es el modelo que ocupa más memoria y tarda más en calcular la segmentación y da valores de IoU más bajos que LinkNet34.

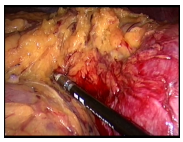
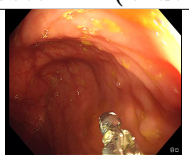
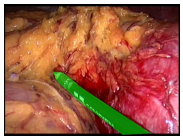
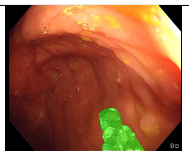
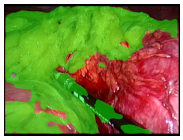
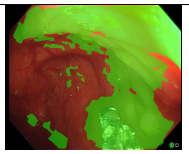
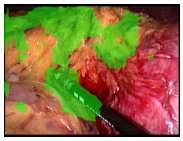
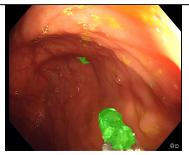
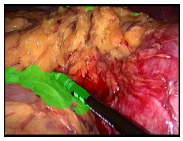
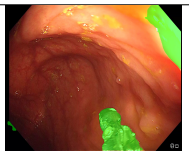
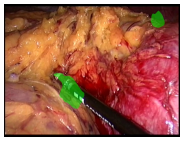
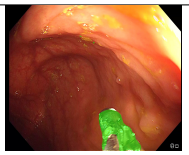
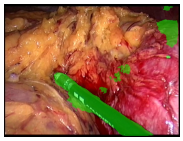
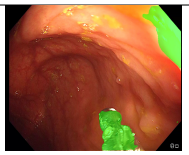
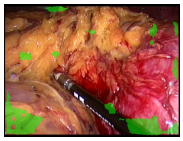
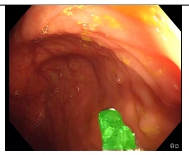
	Dataset Endo15	**Dataset EM (Vídeo E)**
Imagen original		
Segmentación manual		
U-Net		
EM-U-Net		
TernausNet-11		
EM-TernausNet-11		
LinkNet34		
EM-LinkNet34		

Tabla 5.4: Ejemplos de segmentación de herramientas con los modelos del primer experimento. La zona verde corresponde a la parte etiquetada como herramienta por el modelo.

5.2.2. Comparación de distintos modelos

Objetivo. El objetivo de este segundo experimento es mejorar la eficiencia de los modelos entrenados, tanto en precisión como en memoria y tiempo de inferencia.

Descripción. El modelo LinkNet34 permite obtener segmentaciones correctas pero con valores de IoU más bajos que los dados en [11]. Para intentar llegar a valores parecidos, se puede hacer otro *fine-tuning* de este modelo utilizando más datos del proyecto. Otra manera de intentar llegar a un modelo más preciso es entrenar de cero un modelo de segmentación con resultados buenos en imágenes diferentes de las del proyecto. En el caso de este trabajo, se entrena de cero el modelo MiniNet.

El primer paso realizado para entrenar de cero el modelo MiniNet es pre-entrenarlo con los datos del dataset Endo15-A, para tenerlo pre-entrenado con datos parecidos a los utilizados para entrenar LinkNet34 en [11], y así poder comparar los resultados de los dos modelos de manera más justa. Este primer entrenamiento se hace con 50 épocas, batch size de 4 y learning rate de 0,0001. Se ha utilizado la implementación oficial del modelo MiniNet, disponible en el repositorio [26].

Una vez hecho el pre-entrenamiento de MiniNet, se puede hacer el *fine-tuning* de ambos modelos. Para cada uno de los modelos pre-entrenados, se realiza el fine-tuning con datos del dataset EM con dos variaciones :

- Fine-tuning con 3 vídeos (A, B y C) y evaluación con 2 (D y E)
- Fine-tuning con 2 vídeos (A y C) y evaluación con 3 (B, D y E)

Se conservan siempre los vídeos A y C en el conjunto de *train*, ya que el A es el más largo (con más imágenes) y en el C aparece una herramienta que solo se puede ver en este vídeo. Se probó a estudiar el efecto de incluir el vídeo B en el conjunto de *train* o no porque en este vídeo aparecen varias herramientas, de las cuales unas ya aparecen en el vídeo A y otras no, lo que podría cambiar la precisión del modelo entrenado, al incluir en el entrenamiento más tipos de herramientas.

Las dos variaciones de LinkNet se entrenan con 20 épocas, batch size de 8 y learning rate de 0,0001. Las dos variaciones de MiniNet se entrenan con 30 épocas, batch size de 4 y learning rate de 0,0001.

Resultados. La Tabla 5.5 recopila los valores de IoU obtenidos para cada uno de los modelos. En esta tabla, se nota primero que el modelo MiniNet entrenado con los datos del dataset Endo15-A llega a valores de IoU parecidos a los de LinkNet34 en imágenes del dataset Endo15-B pero muy inferiores en imágenes del dataset EM. Una vez realizado el *fine-tuning* de MiniNet, los valores de IoU obtenidos llegan a ser más altos para los modelos de la configuración 2 que para los de la configuración 1, sobre todo en el caso de Mininet evaluado con el vídeo E. Esto puede ser debido a la baja nitidez de las imágenes obtenidas del vídeo B y a la presencia de más brillos y colores diferentes. Por eso, la mejor configuración parece ser la 2. Estos resultados se pueden comprobar también cualitativamente en la figura 5.2.

Modelo	**Trained on**	**Fine-tuned on EM**	**Eval. on Endo15-B**	**Eval on dataset EM**		
				B	**D**	**E**
LinkNet34	Endo17	No	57.36	19.38	26.50	40.59
Mininet	Endo15-A	No	63.43	11.62	13.03	18.21
EM-LinkNet34-1	Endo17	A+B+C	15.13	N/A	60.66	82.94
EM-Mininet-1	Endo15-A	A+B+C	05.13	N/A	70.01	71.37
EM-LinkNet34-2	Endo17	A+C	24.11	49.82	64.23	87.57
EM-Mininet-2	Endo15-A	A+C	10.05	43.06	70.13	88.50

Tabla 5.5: Valores de IoU (%) obtenidos con los dos modelos evaluados. Los modelos marcasos EM-*-# son los obtenidos después del fine-tuning con datos del proyecto (dataset EM) y el número # indica la configuración utilizada para el *fine-tuning*

Para elegir un modelo entre los dos disponibles de la configuración 2, se pueden comparar otras dos métricas, que son el tiempo de inferencia y el número de parámetros, para determinar cual es el modelo más eficiente. Los resultados de estas dos métricas se presentan en la tabla 5.6.

Modelo	**Tiempo de inferencia medio (ms)**	**Número de parámetros (Millones)**
LinkNet34	95	21.79
MiniNet	73	0.52

Tabla 5.6: Tiempo de inferencia medio en el dataset EM y número de parámetros de los dos modelos entrenados.

En esta tabla aparece que MiniNet es más rápido y tiene mucho menos parámetros que LinkNet34, por lo que parece ser el mejor modelo de los dos disponibles para usarlo en tiempo real, ya que, además de dar resultados de IoU parecidos al estado del arte, es también el más eficiente en términos de tiempo de inferencia y memoria.

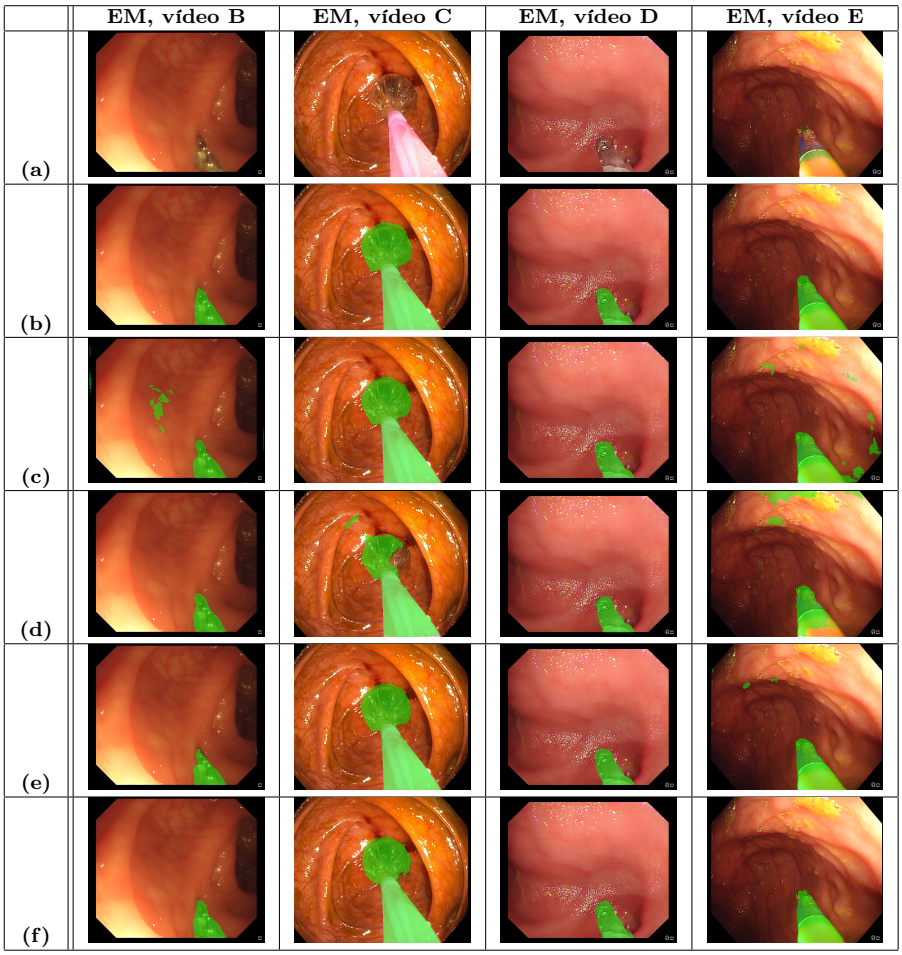


Figura 5.2: Ejemplos de segmentaciones obtenidas en vídeos del dataset EM, con los distintos modelos evaluados en más detalle en el segundo experimento: (a) Imagen original (b) Segmentación manual (c) EM-LinkNet34-1 (d) EM-Mininet-1 (e) EM-LinkNet34-2 (f) EM-Mininet-2

5.3. Análisis no supervisado del contenido de vídeos de endoscopia

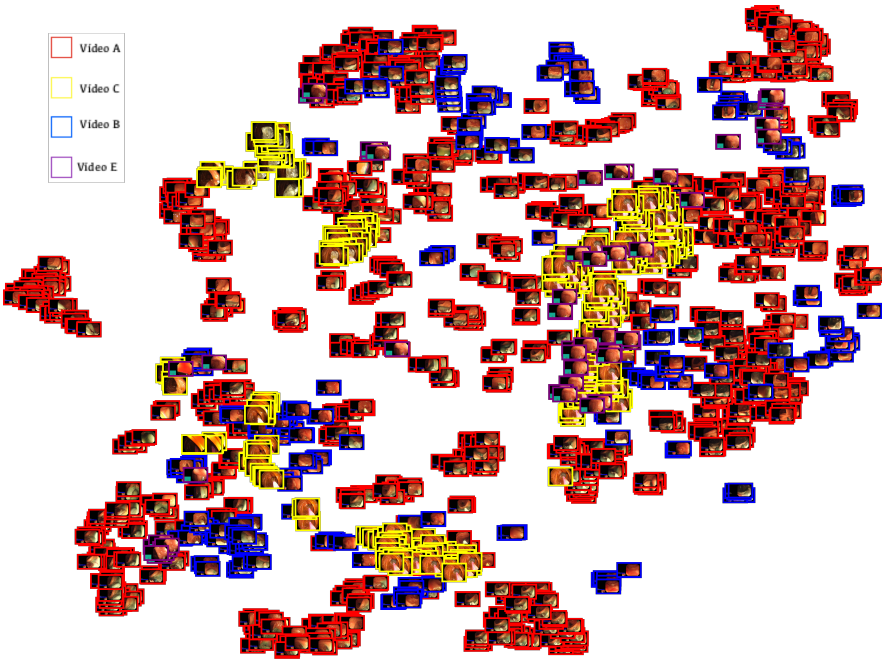
En esta parte se quiere comprobar si las *features* aprendidas por los modelos de segmentación adaptados al proyecto pueden servir para otra tarea de clasificación o anotación semántica de los frames según alguna característica en particular.

5.3.1. Visualización de los frames agrupados por tipo de herramienta

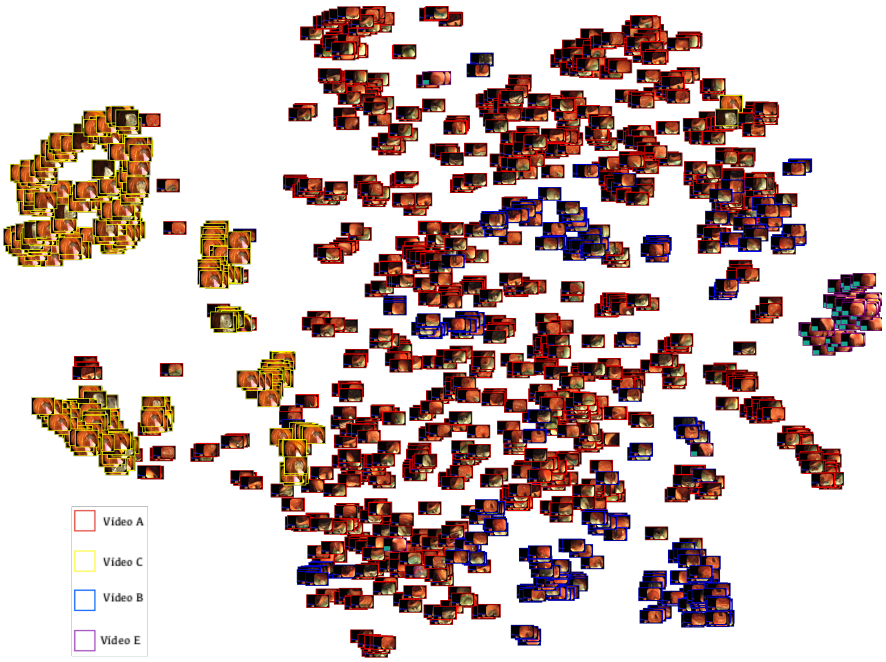
Objetivo. Este primer experimento tiene como objetivo comprobar si el análisis no supervisado de los datos basado en los modelos de segmentación ya entrenados permite distinguir entre los frames según el tipo de herramienta que aparece en cada uno de ellos. Se debería conseguir entonces un grupo de frames para cada tipo de herramienta que pueda aparecer.

Descripción. Para saber si es posible distinguir entre las herramientas, se hace análisis automático de los frames con herramientas de los vídeos A, B, C y E del dataset EM juntos. En efecto, en cada uno de los vídeos aparecen herramientas distintas, con la excepción de los vídeos A y B en los que las herramientas son similares. De esta forma, se podrá saber si el método puede agrupar los frames según el tipo de herramienta que aparece en éste si agrupa los frames que pertenecen al mismo vídeo. Si uno de los modelos de segmentación es adecuado para esta tarea, se deberían visualizar 3 grupos de frames : uno para el vídeo C, uno para el vídeo E y uno para los vídeos A y B.

Resultados. La extracción de los mapas de características se hace utilizando la mejor versión de LinkNet34 y MiniNet obtenida en el apartado anterior (EM-LinkNet34-2 y EM-Mininet-2). El método se aplica a 1 de cada 10 frames ya que se puede suponer que hay poca variación entre 10 frames consecutivos. Esto permite reducir el tiempo que se tarda en extraer las características de todos los frames. Una vez extraídos los mapas de características de cada frame, se aplica el PCA para conservar solo las 50 componentes principales, a las que se aplica luego t-SNE con $perplexity = 20$ y 2000 iteraciones. Las visualizaciones obtenidas se pueden ver en la figura 5.3.



(a)



(b)

Figura 5.3: Visualizaciones obtenidas con los modelos adaptados : (a) EM-LinkNet34-2 y (b) EM-Mininet-2

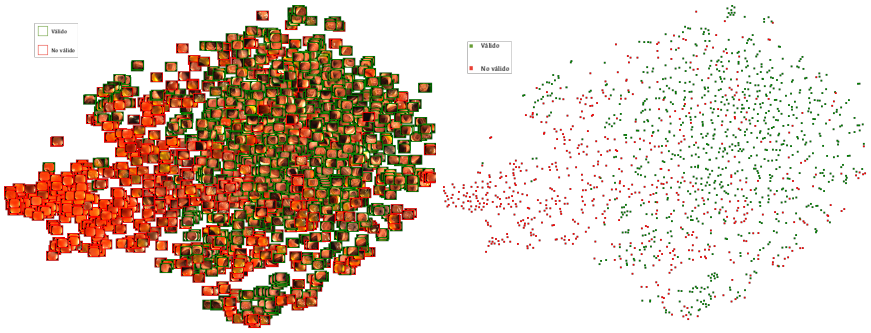
A partir de estas visualizaciones se puede concluir que el análisis automático que utiliza las características extraídas a partir del modelo EM-Mininet-2 si permite separar los frames según el vídeo al que pertenece, y entonces según el tipo de herramienta en el frame. Esto se puede comprobar sobre todo con el vídeo C, que está más alejado del resto de frames ya que contiene una herramienta que solo aparece en este vídeo y no se parece a ninguna de las otras herramientas. Por lo contrario, el análisis automático que utiliza las características extraídas a partir del modelo EM-LinkNet34-2 no separa los frames de los distintos vídeos, por lo que podemos decir que este descriptor no es el más adecuado para esta tarea en particular.

5.3.2. Visualización de los frames informativos y no informativos

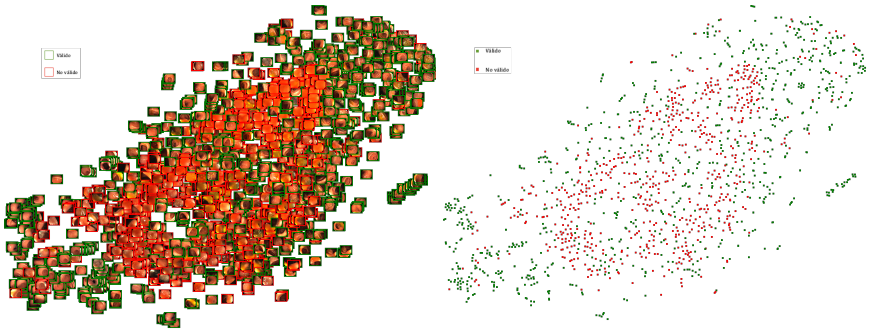
Objetivo. El objetivo de este segundo experimento es comprobar si el análisis automático del vídeo puede permitir separar los frames entre informativos y no informativos, es decir según si aportan o no información relevante para el resto del proyecto.

Descripción. El método se aplica en esta parte a otro vídeo del dataset EM, no utilizado en la parte anterior de segmentación ya que no contiene herramientas pero capturado de la misma manera que el vídeo C. Para comprobar si realmente se agrupan los datos en informativos y no informativos, se dispone para cada frame de la anotación de clase real, es decir una etiqueta asignada manualmente que define si el frame considerado es informativo o no.

Resultados. Al igual que en el experimento anterior, se utiliza 1 de cada 10 frames, los dos mejores modelos de segmentación, PCA con 50 componentes y t-SNE con $perplexity = 20$ y 2000. Las visualizaciones obtenidas se pueden ver en la figura 5.4.



(a)



(b)

Figura 5.4: Visualizaciones obtenidas con los modelos adaptados : (a) EM-LinkNet34-2 y (b) EM-Mininet-2

En estas visualizaciones se puede comprobar que los dos modelos parecen agrupar bastante bien los frames según si son de interés o no. Destaca que la distinción entre los dos grupos parece más clara en el caso de EM-Mininet-2, ya que la división se puede hacer de forma más o menos lineal, mientras que en el caso de EM-LinkNet34-2 los frames informativos se encuentran alrededor de los frames no informativos. Esta repartición podría influir en los resultados obtenidos si se hace a partir de los descriptores extraídos una etapa de *clustering*.

Capítulo 6

Conclusión

6.1. Conclusiones del Trabajo

Todos los objetivos planteados para este trabajo se han alcanzado. Se ha obtenido un sistema automático de segmentación de herramientas adaptado a las imágenes de endoscopia del proyecto. Los modelos de segmentación obtenidos después del *fine-tuning* hecho durante este trabajo son efectivamente mejor adaptados a los datos del proyecto que los modelos originales del estado del arte. También es de notar que se podrían utilizar en tiempo real ya que son bastante eficientes en términos de tiempo de inferencia y memoria ocupada, sobre todo MiniNet.

El segundo objetivo era explorar un sistema no supervisado que permita analizar los datos de las endoscopias para investigar si contienen algún patrón de interés que se pueda detectar con ciertos descriptores. Las técnicas exploradas para visualización muestran que los modelos adaptados han aprendido descriptores de imagen que también pueden ser adecuados para agrupar los frames por distintos contenidos semánticos.

6.2. Principales retos prácticos encontrados

Los principales retos encontrados durante este trabajo vienen de la gran cantidad de datos usados.

En primer lugar, ya que se han usado varias máquinas para llevar a cabo el trabajo, ha sido necesario transferir todos los datos de una a otra, lo que implica mucho tiempo.

Además, para la primera parte de segmentación de herramientas, se ha tenido que segmentar manualmente cerca de 17000 imágenes para poder hacer el entrenamiento de los modelos supervisados. Esto implica también mucho tiempo de trabajo.

Finalmente, varios entrenamientos han sido necesarios para ajustar los parámetros de entrenamiento como el número de épocas o el *learning rate* y así obtener los mejores resultados posibles. Estos entrenamientos son largos (duran días) por el alto número de datos de entrenamiento necesario para obtener buenos resultados.

6.3. Trabajo Futuro

El trabajo realizado se podría extender de diversas maneras.

Primero, en el caso de los modelos de segmentación de herramientas, para mejorar aún más los resultados obtenidos, se podrían obtener más segmentaciones manuales de herramientas, para poder hacer otro *fine-tuning* con aún más datos de *ground truth*.

En el caso del análisis automático de vídeos de endoscopia, se podrían desarrollar modelos de clasificación o de *clustering* no supervisados basados en los modelos de segmentación para poder separar los datos, ya que se ha visto en las visualizaciones obtenidas que las características permitían agrupar los frames según algún criterio que puede ser el tipo de herramienta o si el frame es informativo o no.

Bibliografía

[1] Alexander Rakhlin, Alexey Shvets, Vladimir Iglovikov, and Alexandr A. Kalinin. Deep convolutional neural networks for breast cancer histology image analysis. *Image Analysis and Recognition*, page 737–744, 2018. 1

[2] Aleksei Tiulpin, Jérôme Thevenot, Esa Rahtu, Petri Lehenkari, and Simo Saarakkala. Automatic knee osteoarthritis diagnosis from plain radiographs: a deep learning-based approach. *Scientific reports*, 8(1):1–10, 2018. 1

[3] Aji Resindra Widya, Yusuke Monno, Kosuke Imahori, Masatoshi Okutomi, Sho Suzuki, Takuji Gotoda, and Kenji Miki. 3d reconstruction of whole stomach from endoscope video using structure-from-motion. *arXiv e-prints*, pages arXiv–1905, 2019. 1

[4] EndoMapper. `https://sites.google.com/unizar.es/endomapper`, Abril 2021. 3

[5] Joonmyeong Choi, Keewon Shin, Jinhoon Jung, Hyun-Jin Bae, Do Hoon Kim, Jeong-Sik Byeon, and Namku Kim. Convolutional neural network technology in endoscopic imaging: artificial intelligence for endoscopy. *Clinical endoscopy*, 53(2):117, 2020. 5

[6] Hirotoshi Takiyama, Tsuyoshi Ozawa, Soichiro Ishihara, Mitsuhiro Fujishiro, Satoki Shichijo, Shuhei Nomura, Motoi Miura, and Tomohiro Tada. Automatic anatomical classification of esophagogastroduodenoscopy images using deep convolutional neural networks. *Scientific reports*, 8(1):1–8, 2018. 5

[7] Toshiaki Hirasawa, Kazuharu Aoyama, Tetsuya Tanimoto, Soichiro Ishihara, Satoki Shichijo, Tsuyoshi Ozawa, Tatsuya Ohnishi, Mitsuhiro Fujishiro, Keigo Matsuo, Junko Fujisaki, et al. Application of artificial intelligence using a convolutional neural network for detecting gastric cancer in endoscopic images. *Gastric Cancer*, 21(4):653–660, 2018. 5

[8] Ruikai Zhang, Yali Zheng, Tony Wing Chung Mak, Ruoxi Yu, Sunny H Wong, James YW Lau, and Carmen CY Poon. Automatic detection and classification of colorectal polyps by transferring low-level cnn features from nonmedical domain. *IEEE journal of biomedical and health informatics*, 21(1):41–47, 2016. 5

[9] Bernd Münzer, Klaus Schoeffmann, and Laszlo Böszörmenyi. Content-based processing and analysis of endoscopic images and videos: A survey. *Multimedia Tools and Applications*, 77(1):1323–1362, 2018. 5

[10] Max Allan, Alex Shvets, Thomas Kurmann, Zichen Zhang, Rahul Duggal, Yun-Hsuan Su, Nicola Rieke, Iro Laina, Niveditha Kalavakonda, Sebastian Bodenstedt, et al. 2017 robotic instrument segmentation challenge. *arXiv preprint arXiv:1902.06426*, 2019. 6, 8, 17

[11] Alexey A Shvets, Alexander Rakhlin, Alexandr A Kalinin, and Vladimir I Iglovikov. Automatic instrument segmentation in robot-assisted surgery using deep learning. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 624–628. IEEE, 2018. 6, 8, 10, 11, 17, 20, 23

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 6, 8, 9

[13] Vladimir Iglovikov and Alexey Shvets. Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation. *arXiv preprint arXiv:1801.05746*, 2018. 6, 9

[14] Abhishek Chaurasia and Eugenio Culurciello. Linknet: Exploiting encoder representations for efficient semantic segmentation. In *2017 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4. IEEE, 2017. 6, 10

[15] Alexandre Briot, Prashanth Viswanath, and Senthil Yogamani. Analysis of efficient cnn design techniques for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 663–672, 2018. 6

[16] Iñigo Alonso, Luis Riazuelo, and Ana C Murillo. Mininet: An efficient semantic segmentation convnet for real-time robotic applications. *IEEE Transactions on Robotics*, 36(4):1340–1347, 2020. 6, 10

[17] Wenju Du, Nini Rao, Dingyun Liu, Hongxiu Jiang, Chengsi Luo, Zhengwen Li, Tao Gan, and Bing Zeng. Review on the applications of deep learning in the analysis of gastrointestinal endoscopy images. *Ieee Access*, 7:142053–142069, 2019. 7

[18] Ryoma Bise, Kentaro Abe, Hideaki Hayashi, Kiyohito Tanaka, and Seiichi Uchida. Efficient soft-constrained clustering for group-based labeling. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 421–430. Springer, 2019. 7

[19] Jamil Ahmad, Khan Muhammad, Mi Young Lee, and Sung Wook Baik. Endoscopic image classification and retrieval using clustered convolutional features. *Journal of medical systems*, 41(12):1–12, 2017. 7

[20] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 7, 15

[21] Joris Guérin, Olivier Gibaru, Stéphane Thiery, and Eric Nyiri. Cnn features are also great at unsupervised classification. *arXiv preprint arXiv:1707.01700*, 2017. 14

[22] Carlos Oscar Sánchez Sorzano, Javier Vargas, and A Pascual Montano. A survey of dimensionality reduction techniques. *arXiv preprint arXiv:1403.2877*, 2014. 15

[23] EndoVis Challenge. Endovis'15 instrument subchallenge dataset, `https://opencas.webarchiv.kit.edu/?q=node/30`, 2015. 17

[24] Odin Vision. `https://odin-vision.com`, 2021. 18

[25] Google Colaboratory. `https://colab.research.google.com/notebooks/intro.ipynb`, 2021. 18

[26] MiniNet-v2. `https://github.com/Shathe/MiniNet-v2`, 2021. 23